

REMARKS

[01] Applicant would like to express his appreciation to Examiner Wei Y. Zhen for her time and consideration in discussing the parent application with the undersigned attorney by phone on September 9, 2003. The Interview Summary mailed September 10, 2003 is deemed accurate, except that the undersigned attorney said he would file a continuation, not an RCE. While no specific agreements were sought or reached, the productive exchange of views that was achieved should help bring this application to a proper resolution on the merits.

[02] The Final Office Action in the parent application rejected all pending claims 1-9 as obvious in view of a combination of U.S. Patent No. 6,192,365 to Draper et al., "Draper" herein, and U.S. Patent No. 5,960,189 to Stupek, Jr. et al., "Stupek" et al. The rejections of all claims are traversed.

[03] As discussed in the above-mentioned phone conversation, Applicant has three main grounds of traversal: 1) the objective proposed in the final rejection as a motivation for combining references fails because it is already achieved in the prior art without combining references; 2) one skilled in the art would not apply Draper to program updates because the database related problems that Draper addresses have no counterpart recognized in the prior art; and 3) even if Draper and Stupek were combined as proposed in the final rejection, the result would not yield the claimed invention.

[04] Failure of Proposed Motivation

[05] An obviousness rejection based on a combination of references requires a rationale. It is not enough that the various claim elements be found distributed among plural prior art references. The Final Office Action proposes that one skilled in the art at the time the invention was made would have been motivated to combine Draper and Stupek to “provide an efficient method to keep a current and accurate record of the software program update information to help user to determine whether to perform an upgrade to the software program.” However, the objective of helping a user to determine whether to perform an upgrade is already met by Stupek alone (see Stupek, Abstract “aid the user in determining, whether to perform an upgrade”). One skilled in the art would not be motivated to combine teachings from disparate references to achieve an objective already achieved without combining the references. Hence, the obviousness rejections fail.

[06] Perhaps, the argument is that, despite the fact that Stupek achieves the objective of aiding a user in determining whether to upgrade, the combination of references would allow the objective to be achieved more efficiently. However, there is nothing in either reference that suggests the combination would be more efficient than Stupek alone. In fact, in view of the complexity of the Draper's method relative to that of Stupek's method, one skilled in the art would be unlikely to assume that the combination would be more efficient than the prior-art method.

[07] Since Stupek already teaches a method for efficiently aiding a user in determining whether to update a software program, and since there is no suggestion in either reference that a combination of Stupek with Draper would accomplish this goal any more effectively or efficiently, one skilled in the art would not be motivated to combine the references as proposed in the Final Office Action. Hence, the rejections for obviousness fail.

[08] In the phone interview, the Examiner pointed out that Stupek alone does not render the claims obvious, so that Draper is needed to meet the claim limitations. Applicant does not contest this. However, a desire to reject a claim is not a legitimate motivation for combining references. Since Stupek alone does not render the claims obvious and since there is no legitimate motivation for combining Stupek and Draper, the claims should be allowed.

[09] Inapplicability of Draper's Method to Program Updates

[10] Draper discloses a "reconciliation" ("synchronization") method for reconciling inconsistent database replicas. The program does not rely on advanced knowledge of the correct data or on advanced knowledge of which replica holds the correct data, but does assume that a more recent update takes precedence over a prior update in resolving inconsistencies. The method involves, for each replica, logging each update so that each update is, in effect, time stamped. Reconciliation begins with merging transaction logs chronologically. Optionally, redundant and superceded transactions can be discarded from the log. The resulting merged log can be used to determine a "correct" replica to which all (connected) replicas can be reconciled with minimal user intervention.

[11] The Final Office Action relates Draper's transaction logs and their management to the claimed set of databases and to a claimed generated chronology. One skilled in the art would recognize that Draper's uses transaction logs to resolve inconsistencies among database replicas. This point is illustrated by the following example where the database in question is a list of contacts with their phone numbers.

[12] The person managing the contact list makes updates on a desktop computer at work. The contact manager then copies the contact list to a laptop computer, and makes subsequent updates at home. Forgetting to bring the laptop, the contact manager returns to work the next day, making further updates to the contact list replica on the desktop. That night, further updates are made to the laptop. The following day, the laptop is networked with the desktop. A reconciliation program running on the desktop attempts to reconcile the two contact-list replicas.

[13] The reconciliation program finds an inconsistency between the contact-list replicas, as both list "Jane Doe" as a contact, but the associated phone numbers are different. There is no way to determine which phone number is "correct" by looking at the contact lists alone. However, each computer has maintained a respective transaction log, indicates what updates have been applied and when.

[14] The reconciliation program accesses these logs to identify any changes to Jane Doe's phone number. If the transaction logs indicate that her phone number has been changed on both computers, then the more recent change yields the correct phone number. If the more recent change was performed on the contact-list replica on the laptop, the reconciliation program changes Jane Doe's phone number as indicated by the desktop contact-list replica to conform to that represented in the laptop contact-list replica.

[15] As this example should make clear, the transaction logs are used to resolve inconsistency by update recency. If update recency were not useful in resolving inconsistencies, Draper's method would fail.

[16] In the case of program updates (as opposed to database data updates), update recency is not a useful basis for resolving inconsistencies. In the first place, the time of a program update is not good indicator of anything. It is quite possible, for example, to install an earlier update on one computer at a later time than a later update on another computer. Using a recency criterion to "reconcile" program versions across computers could result in restoring many computers to older rather than newer software. Secondly, program updates are not typically reconciled with each other, but with a common target update. The "correct" update is a given, not an unknown (as it is with inconsistent database replicas). Hence there is no need to resolve inconsistencies among program updates by resorting to transaction logs in the manner taught by Draper.

[17] Since they would recognize that Draper's database reconciliation method is tailored to aspects of database replicas that do not apply to program updates, those skilled in the art would not be motivated to apply Draper's teachings to program updates as proposed in the Final Office Action. Accordingly, for this reason also, the rejections should be withdrawn.

[18] Failure of Combined Teachings to Yield the Claimed Invention

[19] Claim 1 recites both a set of databases and a chronology. The chronology is generated by accessing records of the databases. Clearly, this implies the databases and the chronology are distinct elements. However, the Final Office Action appears to equate both the set of databases and the chronology with Draper's transaction logs, implying they are not distinct elements. But how can a transaction log be generated by accessing its own records? Perhaps, Applicants does not understand the correspondence between claim elements and the elements disclosed by Draper. Applicant requests that the next Office Action be more specific in this regard.

[20] Claim 1 (and others) require that, for each program update, there be a database record that indicates *all* updates it directly supercedes. Draper discloses only that the (*single*) most-recent previous update be indicated in a database record corresponding to a database data update. Since Draper does not disclose the "all" limitation (and since this limitation is not found anywhere in Stupek), the combined teachings could not meet this "all" limitation. Hence, combining Draper and Stupek would not render Claim 1 obvious.

[21] In the phone interview, the Examiner countered this argument by arguing that the single most-recent database data update was the only update directly superceded by a present update. Thus, the argument goes, the single most-recent update in Draper is inherently “all” the directly superceded updates.

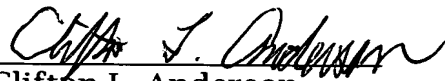
[22] What Draper teaches and what is inherent in Draper’s teachings are two different things. It is a fallacy to include as teachings all things that are inherent in the teachings. For example, teaching that a circle has a 1” diameter is not the same thing as teaching the circle has a π ” circumference, even though the later is inherent in the former. Things that are inherent are only taught if they are made explicit. Draper does not teach that it is valuable to specify “all” directly superceded updates, only that it is valuable to specify the most-recent previous update.

[23] What Draper *teaches* is a field that indicates the most-recent previous update. Applying this teaching to program updates would yield a field that indicates the most-recent previous program update. Since program updates can directly supercede more than one previous program update, it would not be inherent in the context of program updates that “all” program updates would be indicated in a record corresponding to a program update of interest. Accordingly, combining the teachings Draper and Stupek would not meet the “all” limitation of Claim 1.

[24] CONCLUSION

[25] The obviousness rejections fail for three reasons. 1) The proposed motivation for combining references fails as it is achieved in the prior art without combining references. 2) Those skilled in the art would recognize that the use of transaction logs as taught by Draper would not be required or useful in the context of program updates. 3) Combining the references as proposed by the Final Office Action would not yield the claimed invention. Accordingly, it is respectfully submitted that the present application is in condition for allowance, which allowance is respectfully requested.

Respectfully submitted


Clifton L. Anderson
Reg. No. 30,989
(408) 245-0820

New Subtitle and First Paragraph

Cross-Reference to Related Applications

This is a continuation of copending application number 09/732,392, filed on December 7, 2000, which is hereby incorporated by reference herein.